

PHP

PHP : ***H**ypertext **P**reprocessor*⁴, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre⁵, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP⁴, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.⁶ Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

Sommaire

Histoire

Utilisation

Versions

Historique des versions

PHP 6 et Unicode

PHP 7

Présentation

Fonctionnement

Hello world

Balises

Instructions

Structures de contrôle

Génération de code HTML

Programmation orientée objet

Mascotte

Exemples d'application

Plates-formes

Accélération

Notes et références

Voir aussi

Bibliographie

Articles connexes

Liens externes

PHP	
	
Date de première version	1994
Paradigmes	Impératif, orienté objet, fonctionnel, procédural, réflexif et interprété
Auteur	Rasmus Lerdorf
Développeurs	The PHP Group
Dernière version	7.2.4 (28 mars 2018) ¹
Typage	Dynamique, Faible
Influencé par	C/C++ Java Perl ²
Écrit en	C
Système d'exploitation	Multi-plateforme
Licence	Licence libre : Licence PHP ³
Site web	php.net
Extension de fichier	php, phtml, php4, php3, php5, phps et phar

Histoire

Le langage PHP fut créé en 1994 par Rasmus Lerdorf pour son site web. C'était à l'origine une bibliothèque logicielle en C⁷ dont il se servait pour conserver une trace des visiteurs qui venaient consulter son CV. Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la bibliothèque en une implémentation capable de communiquer avec des bases de données et de créer des applications dynamiques et simples pour le Web. Rasmus décida alors en 1995 de publier son code, pour que tout le monde puisse l'utiliser et en profiter⁸. PHP s'appelait alors PHP/FI (pour *Personal Home Page Tools/Form Interpreter*). En 1997, deux étudiants, Andi Gutmans et Zeev Suraski, redéveloppèrent le cœur de PHP/FI. Ce travail aboutit un an plus tard à la version 3 de PHP, devenu alors *PHP: Hypertext Preprocessor*. Peu de temps après, Andi Gutmans et Zeev Suraski commencèrent la réécriture du moteur interne de PHP. Ce fut ce nouveau moteur, appelé *Zend Engine* — le mot *Zend* est la contraction de *Zeev* et *Andi* — qui servit de base à la version 4 de PHP⁹.

Utilisation

En 2002, PHP est utilisé par plus de 8 millions de sites Web à travers le monde¹⁰, en 2007 par plus de 20 millions¹¹ et en 2013 par plus de 244 millions¹².

De plus, PHP est devenu le langage de programmation web côté serveur le plus utilisé depuis plusieurs années :

Année	Part de marché
2010	75 %
2013	75 % ¹³
2016	82 % ¹⁴



Répartition des langages de programmation côté serveur des sites web, le 28 avril 2016.

Enfin en 2010, PHP est le langage dont les logiciels open source sont les plus utilisés dans les entreprises, avec 57 % de taux de pénétration¹⁵.

Versions

La version actuelle est la version 7.1.8, sortie le 3 août 2017¹⁶. Elle utilise *Zend Engine 2* et introduit une modélisation objet plus performante, une gestion des erreurs fondée sur le modèle des exceptions, ainsi que des fonctionnalités de gestion pour les entreprises. PHP 5 apporte beaucoup de nouveautés, telles que le support de SQLite ainsi que des moyens de manipuler des fichiers et des structures XML basés sur *libxml2* :

- une API simple nommée SimpleXML ;
- une API Document Object Model assez complète ;
- une interface XPath utilisant les objets DOM et SimpleXML ;
- intégration de libxslt pour les transformations XSLT via l'extension XSL ;
- une bien meilleure gestion des objets par rapport à PHP 4, avec des possibilités qui tendent à se rapprocher de celles de Java.

Depuis juin 2011 et le nouveau processus de livraison de PHP¹⁷, le cycle de livraison de PHP est d'une mise à jour annuelle comportant des changements fonctionnels importants, la durée de vie d'une branche est de 3 ans et trois branches stables sont maintenues. Cela signifie que lorsqu'une nouvelle version de PHP 5.x sort, la version 5.x-3 n'est plus supportée.

Depuis la sortie de PHP 5.6, la configuration est :

- branche de développement : 7.1 ;
- branches encore supportées (corrections de bugs et de la sécurité) : 7.0, 5.6 ;
- toutes les branches < 5.6 ne sont plus supportées.

Il est à noter qu'historiquement, PHP disposait d'une configuration par défaut privilégiant la souplesse à la sécurité (par exemple register globals, qui a été activé par défaut jusqu'à PHP 4.2¹⁸). Cette souplesse a permis à de nombreux développeurs d'apprendre PHP mais le revers de la médaille a été que de nombreuses applications PHP étaient mal sécurisées. Le sujet a bien été pris en main

par le PHPGroup qui a mis en place des configurations par défaut mettant l'accent sur la sécurité. Il en résultait une réputation de langage peu sécurisé,réputation d'insécurité qui n'a plus de raison d'être^[réf. nécessaire].

Historique des versions

Historique des versions

Version majeure	Version mineure	Support	Date de sortie	Notes
1.0	1.0.0	Ancienne version, non supportée	8 juin 1995	Appelé officiellement "Personal Home Page Tools (PHP Tools)", outils pour page personnelle. C'est la première apparition du nom "PHP" ¹⁹ .
2.0	2.0.0	Ancienne version, non supportée	12 novembre 1997	
3.0	3.0.0	Ancienne version, non supportée	6 juin 1998	Passage d'une personne à une équipe de développeurs. Zeev Suraski et Andi Gutmans réécrivent la base de cette version ¹⁹ .
4.0	4.0.0	Ancienne version, non supportée	22 mai 2000	Ajout d'un système d'analyse syntaxique plus avancé appelé le Zend Engine. Ce moteur procède en deux étapes d'analyse puis d'exécution.
	4.1.0	Ancienne version, non supportée	10 décembre 2001	Introduit les <i>superglobals</i> (\$_GET, \$_POST, \$_SESSION, etc.)
	4.2.0	Ancienne version, non supportée	22 avril 2002	Désactive register_globals par défaut. Les données en provenance du réseau ne sont plus insérées directement dans l'espace de nom global, réduisant ainsi les possibilités de failles de sécurité
	4.3.0	Ancienne version, non supportée	27 décembre 2002	Introduit le CLI, en addition au CGI.
	4.4.0	Ancienne version, non supportée	11 juillet 2005	Ajout du manuel (pages accessibles par la commande man) pour les scripts phpize et php-config.
5.0	5.0.0	Ancienne version, non supportée	13 juillet 2004	Zend Engine II avec un nouveau modèle objet ²⁰ .
5.1	5.1.0	Ancienne version, non supportée	24 novembre 2005	Amélioration de performances par l'introduction de variables de compilation dans un moteur PHP repensé ²⁰ .
5.2	5.2.0	Ancienne version, non supportée	2 novembre 2006	Activation par défaut de l'extension «filter» ²⁰ .
5.3	5.3.0	Ancienne version, non supportée	30 juin 2009 ²¹	Support des <u>espaces de noms</u> ; <u>Late Static Bindings</u> (résolution statique à la volée) ; <u>étiquettes de saut (goto limité)</u> ; <u>fermetures (closures)</u> ; <u>Native PHP archives (phar)</u> ; <u>ramasse-miettes de références cycliques</u> ; connexion persistante avec <u>MySQLi</u> , <u>sqlite3</u> ; ajout de fileinfo en tant que remplacement de mime_magic pour un meilleur support de <u>MIME</u> ; raccourci pour l'opérateur ternaire (?:) <u>fonctions anonymes</u> ; nombreuses corrections de bug ²² ...
	5.3.3	Ancienne version, non supportée	22 juillet 2010	Introduit <u>PHP-FPM</u> ²³
5.4	5.4.0	Ancienne version, non supportée	1 ^{er} mars 2012	Support des <u>traits</u> ; syntaxe courte des tableaux introduite ; accès aux index de tableaux en sortie de fonctions ; les fermetures supportent \$this ; <? = toujours disponible ; accès aux attributs d'un objet dès l'instanciation ; ajout syntaxe Class:: {expr}() ; ajout format de nombres binaires ; suivi

				des envois de fichiers via l'extension de session ; serveur web embarqué en CLI ²⁴ dédié au développement ; support de <u>DTrace</u> ; suppression des extensions <u>magic_quotes</u> , <u>safe_mode</u> , <u>register_globals</u> et <u>register_long_array</u> ²⁴ .
	5.4.26	Ancienne version, non supportée	6 mars 2014	Ajout de <u>ldap_modify_batch()</u>
5.5	5.5.0	Ancienne version, non supportée	20 juin 2013	Support des <u>générateurs</u> et des <u>coroutines</u> ; ajout d'une <u>API de hachage</u> des mots de passe ; intégration en <u>standard</u> du système de cache Zend OPcache ²⁵ ; correction de nombreux bugs ; arrêt du support de <u>Windows XP</u> et <u>Windows Server 2003</u> ; mise à jour de la bibliothèque <u>GD</u> ; les fonctions <u>mysql_</u> sont maintenant considérées comme obsolètes (mise en avant de <u>mysqli_</u> et de <u>PDO_MySQL</u>).
	5.5.4	Ancienne version, non supportée	19 septembre 2013	Amélioration de <u>fputcsv()</u> .
	5.5.5	Ancienne version, non supportée	17 octobre 2013	Nouvelle fonction <u>opcache_compile_file()</u> , support du système <u>GNU Hurd</u> , ajout du type <u>MIME application/pdf</u> dans le serveur de développement.
	5.5.6	Ancienne version, non supportée	14 novembre 2013	Amélioration des performances de <u>array_merge()</u> et <u>func_get_args()</u> .
	5.5.7	Ancienne version, non supportée	12 décembre 2013	Ajout du support de <u>getallheaders()</u> et de types <u>MIME</u> dans le serveur de développement, ajout de <u>apache_response_headers()</u> .
	5.5.10	Ancienne version, non supportée	6 mars 2014	Passage à <u>PCRE 8.34</u> . Ajout de <u>ldap_modify_batch()</u>
	5.5.13	Ancienne version, non supportée	29 mai 2014	Ajout des types <u>MIME xsl/xml</u> au serveur de développement intégré.
5.6	5.6.0	Ancienne version, correctifs de sécurité uniquement	28 août 2014	Nouvelle syntaxe pour les <u>fonctions à paramètres variables</u> et <u>dépaquetage de paramètres</u> , <u>importation de fonctions</u> au sein de <u>espaces de noms</u> , intégration du <u>débogueur phpdbg</u> en <u>standard</u> , amélioration de l'extension <u>zip</u> , expressions acceptées pour la définition de constantes, amélioration de la gestion des <u>données POST</u> , support de la <u>surcharge d'opérateurs</u> dans le Zend Engine, nouvelle <u>méthode magique __debugInfo()</u> fournissant des informations de débogage sur les objets, améliorations diverses sur le support de <u>SSL/TLS</u> , ajout de l'opérateur de <u>exponentiation</u> « ** », téléchargements de poids supérieur à 2 Go désormais supportés.
	5.6.32	Ancienne version, correctifs de sécurité uniquement	26 octobre 2017	Correction de 3 bugs (Date, mcrypt, PCRE)
	5.6.33	Ancienne version, correctifs de sécurité uniquement	04 janvier 2018	Correction de 2 bugs (GD, Phar)

7.0	7.0.0	Ancienne version, améliorations et correctifs de sécurité	3 décembre 2015	Performance améliorée grâce au moteur « png » (jusqu'à deux fois plus rapide que PHP 5.6), support homogène des architectures 64 bits, ajout d'exceptions permettant d'intercepter des erreurs fatales, ajout des opérateurs « ?? » (dit « null coalescing (en) ») et « <=> » (dit « combined comparison »), possibilité de déclarer explicitement le type retourné par une fonction ainsi que les types scalaires des arguments, support des classes anonymes ²⁶ ...
	7.0.25	Ancienne version, améliorations et correctifs de sécurité	26 octobre 2017 ²⁷	Correction de 14 bugs.
	7.0.26	Ancienne version, améliorations et correctifs de sécurité	23 novembre 2017 ²⁸	Correction de 18 bugs.
	7.0.27	Ancienne version, améliorations et correctifs de sécurité	04 janvier 2018	Correction de 11 bugs.
	7.0.28	Ancienne version, améliorations et correctifs de sécurité	27 février 2018	Correction de bugs.
7.1	7.1.0	Ancienne version, améliorations et correctifs de sécurité	1 ^{er} décembre 2016	Types nullable (en), type de retour void, pseudo-type iterable, visibilité des constantes de classes, améliorations de list(), catch de plusieurs types d'exception, etc. ²⁹
	7.1.1	Ancienne version, améliorations et correctifs de sécurité	19 janvier 2017 ³⁰	Correction de 33 bugs et optimisation mémoire de mysqlnd.
	7.1.3	Ancienne version, améliorations et correctifs de sécurité	16 mars 2017 ³¹	Correction de 24 bugs.
	7.1.11	Ancienne version, améliorations et correctifs de sécurité	27 octobre 2017 ³²	Correction de 19 bugs.
	7.1.12	Ancienne version, améliorations et correctifs de sécurité	23 novembre 2017 ³³	Correction de 18 bugs.
	7.1.13	Ancienne version, améliorations et correctifs de sécurité	04 janvier 2018	Correction de 20 bugs.
	7.1.14	Ancienne	01 février	Correction de bugs.

		version, améliorations et correctifs de sécurité	2018		
	7.1.15	Version future	Future version de PHP 7.1	2 mars 2018	Correction de bugs et correctif de sécurité. ³⁴
7.2	7.2.0	Version actuelle, améliorations et correctifs de sécurité	30 novembre 2017 ³⁵	Dernière version de PHP 7. Correction de bugs (voir changelog)	
	7.2.1	Version actuelle, améliorations et correctifs de sécurité	4 janvier 2018 ³⁶	Correction de 21 bugs.	
	7.2.2	Version actuelle, améliorations et correctifs de sécurité	1 février 2018 ³⁶	Correction de 25 bugs.	
	7.2.3	Version future	?? ?? 2018 ³⁶	Future version de PHP 7.2.	

PHP 6 et Unicode

En 2005, le projet de faire de PHP un langage fonctionnant d'origine en Unicode a été lancé par Andrei Zmievski, ceci en s'appuyant sur la bibliothèque International Components for Unicode (ICU) et en utilisant UTF-16 pour représenter les chaînes de caractères dans le moteur³⁷.

Étant donné que cela représentait un changement majeur tant dans le fonctionnement du langage que dans le code PHP créé par ses utilisateurs, il fut décidé d'intégrer cela dans une nouvelle version 6.0 avec d'autres fonctionnalités importantes alors en développement³⁸. Toutefois, le manque de développeurs experts en Unicode ainsi que les problèmes de performance résultant de la conversion des chaînes de et vers UTF-16 (rarement utilisé dans un contexte web), ont conduit au report récurrent de la livraison de cette version³⁹. Par conséquent, une version 5.3 fut créée en 2009 intégrant de nombreuses fonctionnalités non liées à Unicode qui était initialement prévues pour la version 6.0, notamment le support des espaces de nommage (namespaces) et des fonctions anonymes. En mars 2010, le projet 6.0 intégrant unicode fut abandonné et la version 5.4 fut préparée afin d'intégrer la plupart des fonctionnalités non liées à l'unicode encore dans la branche 6.0, telles que les traits ou l'extension des closures au modèle objet⁴⁰.

Le projet est depuis passé à un cycle de livraison prévisible (annuel) contenant des avancées significatives mais contenues tout en préservant au maximum la rétro-compatibilité avec le code PHP existant (5.4 en 2012, 5.5 en 2013, 5.6 prévue pour l'été 2014). Depuis janvier 2014⁴¹, l'idée d'une nouvelle version majeure introduisant Unicode mais se basant sur UTF-8 (largement devenu depuis le standard du Web pour l'Unicode) et permettant certains changements pouvant casser la rétro-compatibilité avec du code PHP ancien est de nouveau discutée et les RFC sont maintenant triées selon leur implémentation en 5.x (évolutions ne causant pas ou marginalement de cassure de la rétro-compatibilité) ou dans la future version majeure (évolutions majeures du moteur et évolutions impliquant une non-compatibilité ascendante).

PHP 7

Au vu des orientations différentes prises par le langage de celles prévues par PHP 6, une partie des développeurs propose de nommer la version succédant à PHP 5 « PHP 7 » au lieu de « PHP 6 ». Un vote parmi les développeurs valide cette proposition par 58 voix contre 24⁴².

La nouvelle version propose une optimisation du code et, d'après la société Zend, offre des performances dépassant celles de machines virtuelles comme HHVM^{43, 44}. Les benchmarks externes montrent des performances similaires pour HHVM et PHP 7, avec un léger avantage d'HHVM dans la plupart des scénarios⁴⁵.

Présentation

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur. PHP peut également générer d'autres formats comme LaTeX, le SVG, le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tels que IIS. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par un navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multi-plateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

La version 5.3 a introduit de nombreuses fonctions nouvelles : les espaces de noms — un élément fondamental de l'élaboration d'extensions, de bibliothèques et de frameworks structurés, les fonctions anonymes, les fermetures, etc.

Aujourd'hui, près de 80 % des sites web utilisent le langage PHP sous ses différentes versions. Plusieurs développeurs PHP responsables de ces sites utilisent en majorité la version 5.4 (38 %) dans leurs missions quotidiennes⁴⁶.

Le langage PHP fait l'objet depuis plusieurs années maintenant d'un forum organisé par l'AFUP (l'Association Française des Utilisateurs de PHP), où experts de la programmation et du milieu se retrouvent pour échanger autour du PHP et de ses développeurs. Ce forum est par ailleurs un moment important pour ces acteurs car il est synonyme du lancement d'une étude consacrée aux développeurs PHP⁴⁷ réalisée par l'AFUP et le cabinet de recrutement Agence-E.

Fonctionnement

PHP appartient à la grande famille des descendants du C, dont la syntaxe est très proche. En particulier, sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à ceci près que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

Dans une utilisation destinée à l'internet, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page de site web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension `.php`), le serveur appelle l'interprète PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML, mais aussi souvent de feuilles de style en cascade et de JS). Ce contenu est renvoyé au serveur HTTP qui l'envoie finalement au client.

Ce schéma explique ce fonctionnement :



Une étape supplémentaire est souvent ajoutée : celle du dialogue entre PHP et la base de données. Classiquement, PHP ouvre une connexion au serveur de SGBD voulu, lui transmet des requêtes et en récupère le résultat, avant de fermer la connexion.

L'utilisation de PHP en tant que générateur de pages Web dynamiques est la plus répandue, mais il peut aussi être utilisé comme langage de programmation ou de script en ligne de commande sans utiliser de serveur HTTP ni de navigateur. Il permet alors d'utiliser de nombreuses fonctions du langage C et plusieurs autres sans nécessiter de compilation à chaque changement du code source.

Pour réaliser en Linux/UNIX un script PHP exécutable en ligne de commande, il suffit comme en Perl ou en Bash d'insérer dans le code en première ligne le shebang : `#!/usr/bin/php48`. Sous un éditeur de développement comme SciTE, même en Windows, une première ligne `<?php` suffit, si le fichier possède un type `.php`.

Il existe aussi une extension appelée PHP-GTK permettant de créer des applications clientes graphiques sur un ordinateur disposant de la bibliothèque graphique GTK+, ou encore son alternative WinBinder.

PHP possède un grand nombre de fonctions permettant des opérations sur le système de fichiers, exécuter des commandes dans le terminal, la gestion des bases de données, des fonctions de tri et hachage, le traitement de chaînes de caractères, la génération et la modification d'images, des algorithmes de compression..

Le moteur de Wikipédia, MediaWiki, est écrit en PHP et interagit avec une base MySQL ou PostgreSQL

Hello world

Quelques exemples du traditionnel Hello world :

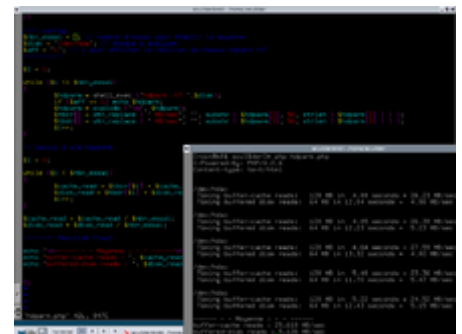
```
<?php
// forme la plus simple, recommandée
echo 'Hello World';
```

`echo` étant une structure du langage, il est possible – et même recommandé – de ne pas mettre de parenthèses, ce qui lui permet en plus d'accepter plusieurs paramètres :

```
<?php
echo 'Hello ', 'World';
// Concaténation avec un point
echo 'Hello ' . 'World';
```

Il est aussi possible d'utiliser la version raccourcie :

```
<?= 'Hello World' ?>
```



Capture d'écran de Vim en fond, affichant un script PHP, et à droite le résultat du script interprété par PHP

Résultat affiché :

```
Hello World
```

Balises

Le code PHP doit être inséré entre des balises `<?php` et `?>`.

Il existe des notations raccourcies : `<?` et `?>`, voire la notation ASP `<%` et `%>`, mais celles-ci sont déconseillées, car elles peuvent être désactivées dans la configuration du serveur (php.ini ou .htaccess) : la portabilité du code est ainsi réduite.

Il existe aussi cette syntaxe, peu courante : `<script language="php">` et `</script>`.

Depuis PHP 7, les notations ASP et script ont été supprimées⁴⁹.

Instructions

Les instructions sont séparées par des ";" (il n'est pas obligatoire après la dernière instruction) et les sauts de ligne ne modifient pas le fonctionnement du programme. Il serait donc possible d'écrire :

```
<?php echo 'Hello World' ; echo 'Comment allez-vous ?' ; echo 'Il fait beau non ?' ?>
```

Pour des raisons de lisibilité, il est néanmoins recommandé d'écrire une seule instruction par ligne. Il est aussi préférable d'écrire le dernier ";".

Structures de contrôle

Le code PHP est composé par des appels à des fonctions, dans le but d'attribuer des valeurs à des variables, le tout encadré dans des conditions, des boucles. Exemple :

```
<?php
// la fonction strtolower renvoie en minuscules la chaîne de caractères passée en paramètre
$lang = strtolower($_POST['lang']);

if ($lang === 'fr')
    echo 'Vous parlez français !' ;
elseif ($lang === 'en')
    echo 'You speak English!' ;
else
    echo 'Je ne vois pas quelle est votre langue !' ;
```

Une condition est appliquée quand l'expression entre parenthèses est évaluée à 'true', et elle ne l'est pas dans le cas de 'false'. Sous forme numérique, 0 représente le 'false', et 1 (et tous les autres nombres) représentent le 'true'.

Le code précédent pourrait aussi être écrit de cette manière :

```
<?php
$lang = strtolower($_POST['lang']);
$is_lang_fr = $lang === 'fr';
$is_lang_en = $lang === 'en';

if ($is_lang_fr)
    echo 'Vous parlez français !' ;
elseif ($is_lang_en)
    echo 'You speak English!' ;
else
    echo 'Je ne vois pas quelle est votre langue !' ;
```

Ici on teste l'égalité entre `$lang` et "fr", mais pas directement dans le if : le test retourne un boolean (c'est-à-dire soit true, soit false) qui est stocké dans la variable `$is_lang_fr`. On entre ensuite cette variable dans le if et celui-ci, selon la valeur de la variable, effectuera ou non le traitement.

Les blocs if, elseif et else sont généralement délimités par les caractères "{" et "}", qui peuvent être omis, comme dans les codes précédents, lorsque ces blocs ne contiennent qu'une instruction.

Il est également possible d'écrire *else if* en deux mots, comme en C/C++.

Génération de code HTML

On peut générer du code HTML avec le script PHP par exemple :

```
<?php
$nom = 'Albert';
echo '<div>Bonjour ' . $nom . ' !</div>';
```

Il est également possible d'utiliser une syntaxe alternative pour la structure if/else :

```
<?php
$nomsAutorises = [ 'Albert', 'Bertrand' ];
$nomEnCours = 'Eve';
?>
<?php if (in_array($nomEnCours, $nomsAutorises)): ?>
    <p>Bonjour <?php echo $nomEnCours; ?> !</p>
<?php else: ?>
    <p>Vous n'êtes pas un utilisateur autorisé !</p>
<?php endif; ?>
```

Une autre approche consiste à concaténer l'intégralité du code HTML dans une variable et de réaliser un echo de la variable en fin de fichier :

```
<?php
$out = '';

// On ajoute progressivement tout le code html
$out .= '<p>a' . 'b' . 'c</p>';
$out .= '<div class="bottom"><p>d' . 'e' . 'f</p></div>';

// On affiche le contenu
echo $out;
```

Dans le cas où l'utilisateur aura préféré l'utilisation de la commande echo à la concaténation, il lui sera possible de capturer le flux en utilisant les fonctions `ob_start()` et `ob_get_clean()`:

```
<?php
// On place le flux dans un tampon
ob_start();

// On fait des affichages
echo 'a', 'b', 'c', "\n";
echo 'd', 'e', 'f', "\n";
require_once 'fichier.php'; // De nombreux affichages (echo)

// On stoppe la mise en tampon, on récupère son contenu et on met le tampon à vide (pour une éventuelle
prochaine temporisation)
$out = ob_get_clean();

// On affiche le contenu du tampon
echo $out;
```

PHP, tout comme JavaScript, permet aussi de construire un modèle objet de document (*DOM*)⁵⁰, ce qui permet de créer ou modifier un document (X)HTML sans écrire de HTML, comme le montre l'exemple suivant :

```
<?php
$doctype = DOMImplementation::createDocumentType('html');
$dom = DOMImplementation::createDocument(null, "html", $doctype);
$html = $dom->createElement();
$html->head = $dom->createElement("head");
$html->appendChild($html->head);
$html->head->title = $dom->createElement("title");
$html->head->title->nodeValue = "Exemple de HTML";
$html->head->appendChild($html->head->title);
$html->head->charset = $dom->createElement("meta");
$html->head->charset->setAttribute("http-equiv", "Content-Type");
$html->head->charset->setAttribute("content", "text/html; charset=utf-8");
$html->head->appendChild($html->head->charset);
$html->body = $dom->createElement("body");
$html->appendChild($html->body);
$html->body->p = $dom->createElement("p");
$html->body->p->nodeValue = "Ceci est un paragraphe.";
$html->body->appendChild($html->body->p);
$html->body->p->br = $dom->createElement("br");
$html->body->p->appendChild($html->body->p->br);
$html->body->p->a = $dom->createElement("a");
$html->body->p->a->nodeValue = "Ceci est un lien.";
$html->body->p->a->setAttribute("href", "cible.html");
$html->body->p->appendChild($html->body->p->a);

echo $dom->saveHTML();
```

Qui crée le code HTML suivant :

```
<!DOCTYPE html>
<html>
<head>
<title>Exemple de HTML</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
</head>
<body><p>Ceci est un paragraphe. <br /><a href="cible.html">Ceci est un lien. </a></p></body>
</html>
```

Cette méthode est cependant peu utilisée pour générer un document complet, on l'utilise généralement pour générer un fichier XML.

La commande `phpinfo()` est aussi utilisée pour générer un code HTML décrivant les paramètres du serveur ; elle est aussi très utilisée pour tester la bonne exécution du moteur d'exécution PHP

Programmation orientée objet

Comme en C++ et en Java, PHP permet de programmer en orienté objet, en créant des classes contenant des attributs et des méthodes, qui peuvent être instanciées ou utilisées en statique.

Toutefois, PHP est un langage à héritage simple⁵¹, c'est-à-dire qu'une classe ne peut hériter que d'au plus une seule autre classe (sinon il faut utiliser un trait pour simuler l'héritage multiple par composition). Cependant les interfaces peuvent en étendre plusieurs autres⁵².

Voici un exemple de création d'une classe :

```
<?php
class Perso {
    const PV_INITIAL = 2000;
    private $pv;

    public function __construct($pv = false) { // Paramètre optionnel
        if (!is_numeric($pv) || $pv < 0 || $pv > 100000000)
            $this->pv = self::PV_INITIAL;
        else
            $this->pv = $pv;
    }
}
```

```

// Accesseurs
public function getPv() {
    return $this->pv;
}

public function isDead() {
    return $this->pv == 0;
}
}

// Création d'une classe enfant de Perso
class Magicien extends Perso {
    private $magie;
}

// Création d'une instance de classe
$perso = new Perso(1000);

// Utilisation de l'objet
echo 'Votre personnage a ' . $perso->getPV() . ' PV.';

// Constantes de classes
echo 'Le PV par défaut attribué à un nouveau personnage est de ' . Perso::PV_INITIAL . '.';

// Destruction de l'objet
unset($perso);

```

Mascotte

Comme de nombreux projets Open Source, PHP possède une mascotte. Il s'agit de l'éléPHPant, dessiné en 1998 par El Roubio⁵³.

El Roubio s'est inspiré de la ressemblance des lettres PHP avec un éléphant et du fait que deux des lettres du langage soient déjà présentes dans ce mot, ce qui a permis de créer le néologisme éléPHPant. Toutes les œuvres d'El Roubio sont distribuées sous licence GNU GPL. Une peluche de l'éléPHPant⁵⁴ bleu existe. D'autres versions ont vu le jour ces dernières années (rose, jaune, rouge, violet et orange) sous l'impulsion de sociétés (PHP Architect ou Zend Technologies) ou de groupes utilisateurs comme PHP Women.



L'éléPHPant, la mascotte de PHP.

Exemples d'application

- Wiki (MediaWiki, WikiNi, DokuWiki...)
- forum (phpBB, Vanilla, IPB, punBB...)
- Systèmes de gestion de blog (Dotclear, WordPress...)
- Systèmes de gestion de contenu (appelés aussi CMS) (Sip, ExpressionEngine, Drupal, Xoops, Joomla...)
- Administration de bases de données (phpMyAdmin, phpPgAdmin, Adminer, Eskuel...)
- Frameworks (Zend Framework, CodeIgniter, CakePHP, Symfony, etc.)
- Logiciel ECM (Dynacase Platform)
- Logiciel BPM (Dynacase Platform)
- E-commerce (Magento, osCommerce, PrestaShop, etc.)

Plates-formes

Un serveur Web en architecture trois tiers est composé d'un système d'exploitation, un serveur HTTP, un langage serveur et enfin un système de gestion de base de données (SGBD), cela constituant une plate-forme.

Dans le cas de PHP comme langage serveur, les combinaisons les plus courantes sont celles d'une plateforme LAMP (pour Linux Apache MySQL PHP) et WAMP (Windows Apache MySQL PHP). Une plate-forme WAMP s'installe généralement par le biais d'un seul logiciel qui intègre Apache, MySQL et PHP, par exemple EasyPHP, VertrigoServ, WampServer ou UwAmp. Il existe le même type de logiciels pour les plates-formes MAMP (Mac OS Apache MySQL PHP), à l'exemple du logiciel MAMP.

Il existe d'autres variantes, par exemple les plates-formes LAPP (le M de MySQL est remplacé par le P de PostgreSQL) ou encore le logiciel XAMPP (Apache MySQL Perl PHP ; le X indique que le logiciel est multiplate-forme), un kit de développement multiplate-forme.

On peut décliner une grande variété d'acronymes sous cette forme. Des confusions peuvent parfois exister entre la plate-forme en elle-même et le logiciel permettant de l'installer, si elles ont le même nom. Il faut également remarquer que la grande majorité des logiciels « tout en un » sont destinés au développement d'applications Web en local, et non à être installés sur des serveurs Web. Une exception à cette règle est peut-être Zend Server, le serveur distribué par Zend Technologies, qui est prévu pour fonctionner aussi bien en environnement de développement que de production.

Accélération

PHP est à la base un langage interprété, ce qui est au détriment de la vitesse d'exécution du code. Sa forte popularité associée à son utilisation sur des sites Web à très fort trafic (Yahoo, Facebook⁵⁵) ont amené un certain nombre de personnes à chercher à améliorer ses performances pour pouvoir servir un plus grand nombre d'utilisateurs de ces sites Web sans nécessiter l'achat de nouveaux serveurs.

La réécriture du cœur de PHP, qui a abouti au Zend Engine pour PHP 4 puis au Zend Engine 2 pour PHP 5, est une optimisation. Le Zend Engine compile en interne le code PHP en bytecode exécuté par une machine virtuelle. Les projets open source APC et eAccelerator fonctionnent en mettant le bytecode produit par Zend Engine en cache afin d'éviter à PHP de charger et d'analyser les scripts à chaque requête. À partir de la version 5.5 de PHP, le langage dispose d'un cache d'opcode natif (appelé OpCache) rendant obsolète le module APC.

Il existe également des projets pour compiler du code PHP :

- Roadsend et phc compilent du PHP en C,
- Quercus compile du PHP en bytecode Java exécutable sur une machine virtuelle Java
- Phalanger compile du PHP en Common Intermediate Language exécutable sur le Common Language Runtime du framework .NET.
- HipHop for PHP transforme du PHP en C++ qui est ensuite compilé en code natif. Ce projet open source a été démarré par Facebook⁵⁶.

Notes et références

1. <https://git.php.net/?p=php-src.git;a=tag;h=refs/tags/php-7.2.4>
2. Manuel PHP – Préface (<http://www.php.net/manual/fr/preface.php>)
3. source (http://www.php.net/license/3_01.txt)
4. « Manuel PHP : Préface » (<http://www.php.net/manual/fr/preface.php>) sur *PHP.net* (<http://www.php.net/>) (consulté le 7 novembre 2007).
5. **(en)** « La licence de PHP est reconnue libre par la *Free Software Foundation* » (http://www.fsf.org/licensing/licenses/index_html#GPLIncompatibleLicenses) (consulté le 7 novembre 2007)
6. « Introduction à PHP » (<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/introduction-a-php#/id/r-2172370>) sur *openclassrooms.com* (consulté le 14 juillet 2015).
7. Rasmus Lerdorf, « I wonder why people keep writing that PHP was ever written in Perl. It never was. #php » (<https://twitter.com/rasmus/status/226405807305138176>) [**html**], Twitter, 20 juillet 2012 (consulté le 5 septembre 2014)
8. <https://groups.google.com/forum/#!msg/comp.infosystems.www.authoring.cgi/PyJ25gZ6z7A/M9FkTUVdfcwJ>
9. « Histoire de PHP » (<http://www.php.net/manual/fr/history.php>), sur *PHP.net* (<http://www.php.net/>) (consulté le 7 novembre 2007).
10. **(en)** « Netcraft web Server Survey - février 2002 » (<http://survey.netcraft.com/index-200202.html>) (consulté le 7 novembre 2007)
11. <http://www.php.net/usage.php>
12. <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>
13. « Server-side programming language statistics » (<http://blog.stonieriverlearning.com/top-5-programming-languages-used-in-web-development/>) 6 mars 2013
14. « Server-side Programming Languages » (<http://w3techs.com/>) w3techs, 28 avril 2016

15. « PHP and Perl crashing the enterprise party» (http://news.cnet.com/8301-13505_3-10453213-16.html)
16. (en) « Historique des sorties de PHP» (<http://www.php.net/releases/>) sur *PHP.net* (<http://www.php.net/>) (consulté le 7 novembre 2007)
17. <https://wiki.php.net/rfc/releaseprocess>
18. « Register Globals» (<http://php.net/manual/en/securityglobals.php>).
19. (en) « History of PHP» (<http://www.php.net/manual/en/historyphp.php>), The PHP Group (consulté le 18 juin 2010)
20. (en) « PHP: PHP 5 ChangeLog» (<http://www.php.net/ChangeLog-5.php>) The PHP Group, 8 novembre 2007 (consulté le 22 février 2008)
21. (en) *PHP 5.3.0 Released!* (<http://www.php.net/archive/2009.php#id2009-06-30-1>)
22. (en) *PHP 5.3.0 Changelog* (<http://php.net/ChangeLog-5.php#5.3.0>)
23. (en) *FPM in 5.3* (https://groups.google.com/group/highload-php-en/browse_thread/thread/8af45714ab362c41?pli=1)
24. [1] (<http://php.net/manual/fr/features.commandline.webserve.php>)
25. (en) *PHP 5.5.0 released.* (<http://php.net/archive/2013.php#id2013-06-20-1>)
26. (en) *PHP 7.0.0 Released* (<http://php.net/archive/2015.php#id2015-12-03-1>)
27. (en) [2] (<http://php.net/ChangeLog-7.php#7.0.25>)
28. (en) [3] (<http://php.net/ChangeLog-7.php#7.0.26>)
29. (en) *PHP 7.1.0 Released* (<http://php.net/archive/2016.php#id2016-12-01-3>)
30. (en) *PHP 7.1.1 Released* (<http://php.net/ChangeLog-7.php#7.1.1>)
31. (en) *PHP 7.1.3 Released* (<http://www.php.net/ChangeLog-7.php#7.1.3>)
32. (en) *Released* (<https://secure.php.net/ChangeLog-7.php#7.1.11>)
33. (en) *Released* (<https://secure.php.net/ChangeLog-7.php#7.1.12>)
34. (en) « PHP: News Archive - 2018» (<http://php.net/archive/2018.php#id2018-03-02-1>) sur *php.net* (consulté le 24 mars 2018)
35. (en) [4] (<http://www.php.net/ChangeLog-7.php#7.2.0>)
36. (en) [5] (<https://wiki.php.net/todo/php72>)
37. Andrei Zmievski, « PHP Unicode support design document (mailing list post)» (<http://marc.info/?l=php-internals&m=112365908921757&w=1>) 10 août 2005 (consulté le 9 février 2014)
38. « PHP 5.5 or 6.0» (<http://news.php.net/php.internals/17668>) (consulté le 9 février 2014)
39. Andrei Zmievski, « The Good, the Bad, and the Ugly: What Happened to Unicode and PHP 6» (<http://www.slideshare.net/andreizm/the-good-the-bad-and-the-ugly-what-happened-to-unicode-and-php-6>) (consulté le 9 février 2014)
40. Rasmus Lerdorf, « PHP 6 (mailing list post)» (<http://news.php.net/php.internals/47120>) 11 mars 2010 (consulté le 7 février 2014)
41. Adam harvey, « Ruminations on PHP 5++ (message de liste de discussion)» (<http://news.php.net/php.internals/71481>), 11 mars 2010 (consulté le 7 février 2014)
42. Andrea Faulds, Zeev Suraski, « PHP RFC: Name of Next Release of PHP» (<https://wiki.php.net/rfc/php6>) 2004 (consulté le 22 novembre 2014)
43. « PHP Next Generation is outperforming Facebook's HHVM » (<http://blog.zend.com/2014/07/17/php-next-generation-outperforming-facebooks-hhvm/>) sur *The Zend Blog*, 17 juillet 2014 (consulté le 22 novembre 2014)
44. Antoine Crochet-Damais, « Andi Gutmans (Zend Technologies) : "PHP Next Generation est plus performant que HHVM de Facebook" » (<http://www.journaldunet.com/developpeur/php/andi-gutmans-interview-de-andi-gutmans.shtml>), sur *Le Journal du Net*, 25 juin 2014 (consulté le 22 novembre 2014)
45. Zeev Suraski, « Benchmarking PHPNG!» (<http://zsuraski.blogspot.be/2014/07/benchmarking-phpng.html>) sur *Zeev's blog*, juillet 2014 (consulté le 22 novembre 2014)
46. Agence-E, « Baromètre AFUP - Agence-E 2015 : Les salaires dans l'écosystème PHP en France» (<http://www.agence-e.fr/images/stories/Barometre-AFUP-Agence-e-2015.pdf>) [PDF], sur *Agence-e.fr*, 25 septembre 2014 (consulté le 11 décembre 2014).
47. « Baromètre AFUP - Agence-e 2015 : les salaires du PHP en France » (<http://www.agence-e.fr/component/content/article/11-sondages/39-barometre-afup-les-salaires-du-php-en-france-2014>) Agence-e.
48. `/usr/bin/` est le répertoire standard des fichiers binaires exécutables sur la plupart des distributions.
49. (en) « PHP RFC: Remove alternative PHP tags» (https://wiki.php.net/rfc/remove_alternative_php_tags)
50. <http://php.net/manual/fr/book.dom.php>
51. Sébastien Rohaut, *Algorithmique : Techniques fondamentales de programmation (avec des exemples en PHP)* Editions ENI, 2009 (lire en ligne (<https://books.google.fr/books?id=IfEft8zQgQcC&pg=PT362&dq=php+%22h%C3%A9ritage+simple%22&hl=fr&sa=X&ved=0ahUKEwjFpuyt76PPhVCqxOKHdiyCwkQ6AEILDAB#v=onepage&q=%22PHP%20autorise%20le%20h%C3%A9ritage%20simple%20mais%20pas%20le%20h%C3%A9ritage%20multiple%22&f=false>))
52. <http://php.net/manual/fr/language.oop5.interfaces.php>





53. Présentation des éléPHPants par El Roubio(<http://www.elroubio.net/?p=elephpant>)
54. <http://www.elephpant.com/>
55. Facebook looks to fix PHP performance with HipHop virtual machine(<https://arstechnica.com/business/2011/12/facebook-looks-to-fix-php-performance-with-hiphop-virtual-machine/>.)
56. HipHop for PHP: Move Fast(<http://developers.facebook.com/news.php?blog=1&story=358>)Haiping Zhao le 2 février 2010

Voir aussi

Bibliographie

- **(en)** Luke Welling, Laura Thomson, *PHP and MySQL Web development*, Sams Publishing, 2008, 4^e éd. (ISBN 0-672-32916-6, OCLC 854795897)
- Damien Seguy, Philippe Gamache, *Sécurité PHP 5 et MySQL, 3^e édition*, Eyrolles, 1^{er} décembre 2011, 277 p. (ISBN 2-212-13339-1)
- Jean Engels *PHP 5 Cours et Exercices, 3^e édition*, Eyrolles 2013, 631 pages (ISBN 978-2-212-13725-5)

Sur les autres projets Wikimedia :

-  [PHP](#), sur Wikimedia Commons
-  [PHP](#), sur le Wiktionnaire  [PHP](#), sur Wikiversity
-  [PHP](#), sur Wikibooks

Articles connexes

- [Paamayim Nekudotayim](#): nom de l'opérateur `::` en PHP
- [Liste de frameworks PHP](#): liste des cadres de développement (Frameworks) en PHP

Liens externes

- [Site officiel](#)
- Manuel officiel PHP, sur le site php.net
- [Entretien avec Rasmus Lerdorf](#) à l'occasion du 10^e Forum PHP (2010)

Ce document provient de «<https://fr.wikipedia.org/w/index.php?title=PHP&oldid=147393992>».

La dernière modification de cette page a été faite le 10 avril 2018 à 15:38.

Droit d'auteur : les textes sont disponibles sous licence Creative Commons attribution, partage dans les mêmes conditions ; d'autres conditions peuvent s'appliquer. Voyez les conditions d'utilisation pour plus de détails, ainsi que les crédits graphiques. En cas de réutilisation des textes de cette page, voyez comment citer les auteurs et mentionner la licence.

Wikipedia® est une marque déposée de la Wikimedia Foundation, Inc, organisation de bienfaisance régie par le paragraphe 501(c)(3) du code fiscal des États-Unis.